A T A R I   400   A N D   800

SERIAL INPUT/OUTPUT PORT

USER'S HANDBOOK

INTRODUCTION

The ATARI personal computers communicate with peripheral devices over a
19,200 baud asynchronous serial port. The serial port consists of a serial
DATA OUT (transmission) line, a serial DATA IN (receiver) line and other
miscellaneous control lines described in the serial I/O Bus Protocol section.
Data is transmitted and received as 8 bits of serial data (LSB sent first)
preceeded by a logic zero start bit and succeeded by a logic true stop bit.
DATA OUT is transmitted as positive logic (+4v = TRUE, Ov = FALSE). The
serial DATA OUT always changes when the serial CLOCK OUT goes true. The
CLOCK OUT then returns to zero in the center of the DATA OUT bit time.

This document details the mechanics of serial port communications. For
information about proper peripheral device operations on a system level,
consult the Colleen I/O Subsystem Manual and the application note "Adding
a Non-Resident Device Handler to the Operating System." For information
about serial communications that don't conform to the 19,200 baud serial
protocol consult the Colleen Hardware Manual.

```
      2   4   6   8   10  12
      0   0   0   0   0   0


      0   0   0   0   0   0   0
      1   3   5   7   9   11  13
```

1. Clock In

2. Clock Out

3. Data In to Computer

4. GND

5. Data Out of Computer

6. GND

7. $\overline{\text{Command}}$

8. Motor Control

9. Proceed

10. +5 / Ready

11. Audio In

12. +12

13. $\overline{\text{Interrupt}}$

1. CLOCK IN – Not required in present serial port protocol. Can be used in future synchronous communication schemes.

2. CLOCK OUT– Serial port clock. DATA OUT always changes when the CLOCK OUT then returns to zero in the center of the DATA OUT bit time.

3. DATA IN – 19200 baud data line to the computer.

4. GND   – Ground

5. DATA OUT – 19200 baud data line from the computer.

6. GND   – Ground

7. $\overline{\text{COMMAND}}$ – Goes to zero when a command frame is being sent.

8. MOTOR CONTROL – Motor control line for the cassette tape recorder.

9. $\overline{\text{PROCEED}}$ – Not implemented in present serial port protocol. This line generates an IRQ interrupt when it goes to zero.

10. +5/READY – Indicates that the computer is turned on and ready. Restricted use as a +5 volt supply.

11. AUDIO IN – Audio signal from the cassette to the R.F. modulator.

12. +12 VOLT –

13. $\overline{\text{INTERRUPT}}$– Not implemented in present serial port protocol. This line generates an IRQ interrupt when it goes to zero.

PERIPHERAL INPUT:

$V1H = 2V.$ min

$V1L = .4V$ max

$I1H = 20$ ua max
@ 2.0 $V1H$

$I1L = 5$ ua max
@ $V1L=.4V.$

Peripheral Output

(Open collector bipolar)

$VoL = .4V.$ max
@1.6 ma.

$VoH = 4.5$ V min
With R ext =
100 K.

Vcc/Ready Input only

$V1H = 2V.$ min
@ $I1H = 1$ ma
max.

$V1L = .4V.$ max

Input goes to logic
"o" when open.

SERIAL I/O BUS PROTOCOL
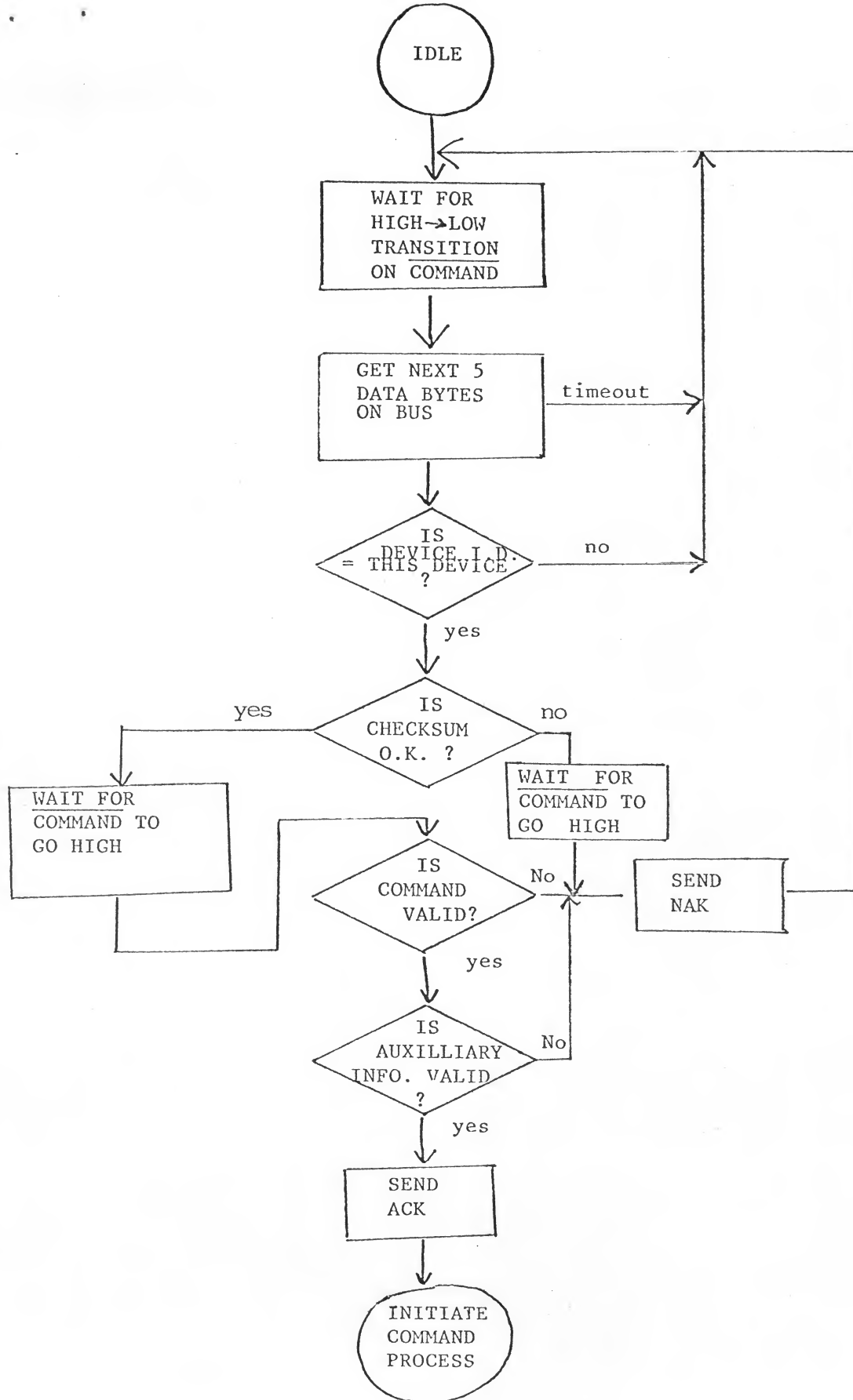
X - 1

2. Bus Command Types

The Serial Bus Protocol accomodates three basic types of
commands: 1) data send, 2) data receive, and 3) immediate
(no data - command only).

The common element in all three command types is the command
frame, five bytes of information sent while the COMMAND line is
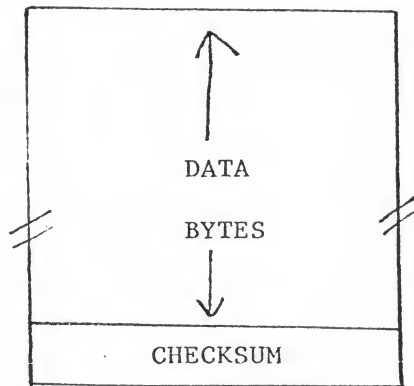held low.

The command frame format is shown below:

```
+-----------------------+
|                       |
|   SERIAL BUS          |
|   DEVICE I.D.         |
|                       |
+-----------------------+
|                       |
|   COMMAND             |
|                       |
+-----------------------+
|   AUXILLIARY          |
|                       |
|   INFORMATION         |
+-----------------------+
|   CHECKSUM            |
+-----------------------+
```

The flowchart on the next page shows a typical device controller's
processing of a command frame.

X - 2

Following the command frame is an optional data frame which
is formatted as shown below:

```
          ┌────────────────────────────┐
          │              ↑             │
          │                            │
          │            DATA            │
        //│                            │//
          │            BYTES           │
          │                            │
          │              ↓             │
          ├────────────────────────────┤
          │          CHECKSUM          │
          └────────────────────────────┘
```
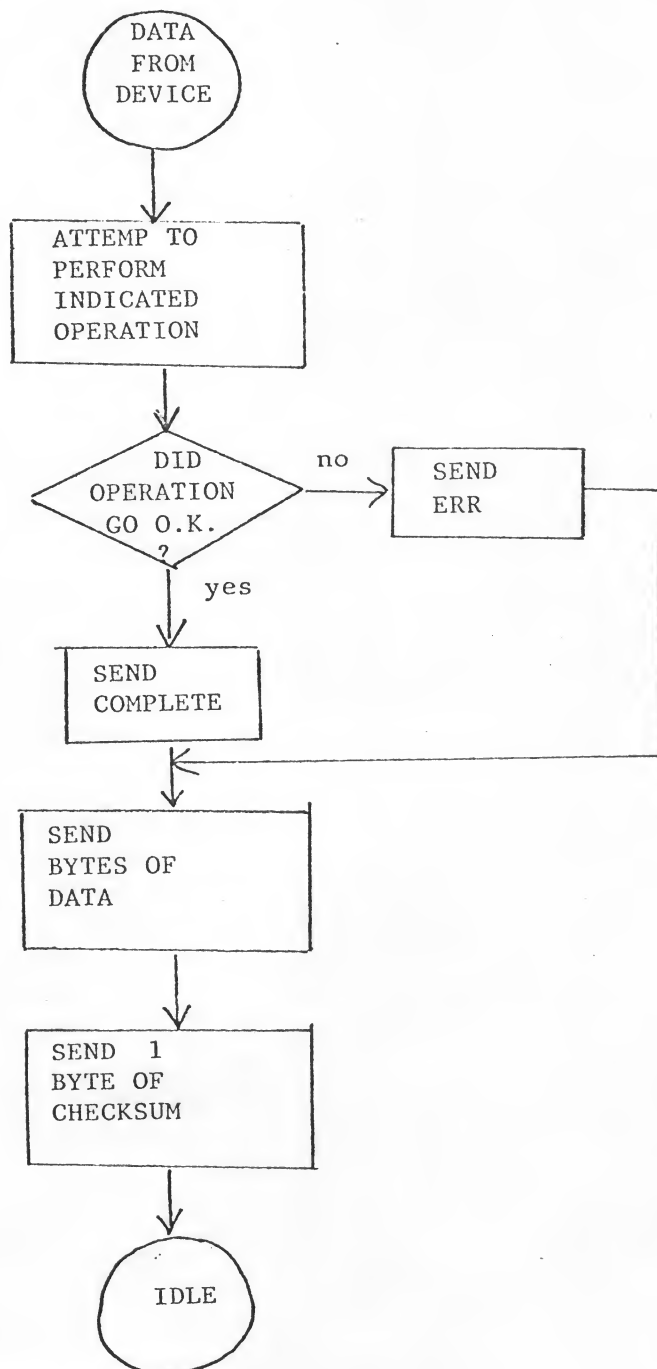
This data frame may originate at the central processor or at the
device, depending upon the command. The flowchart on the next page
shows a typical device controller's processing of a data frame from
the central processor.

See appendix K for actual Serial Bus codes for ack, nak, complete, etc.
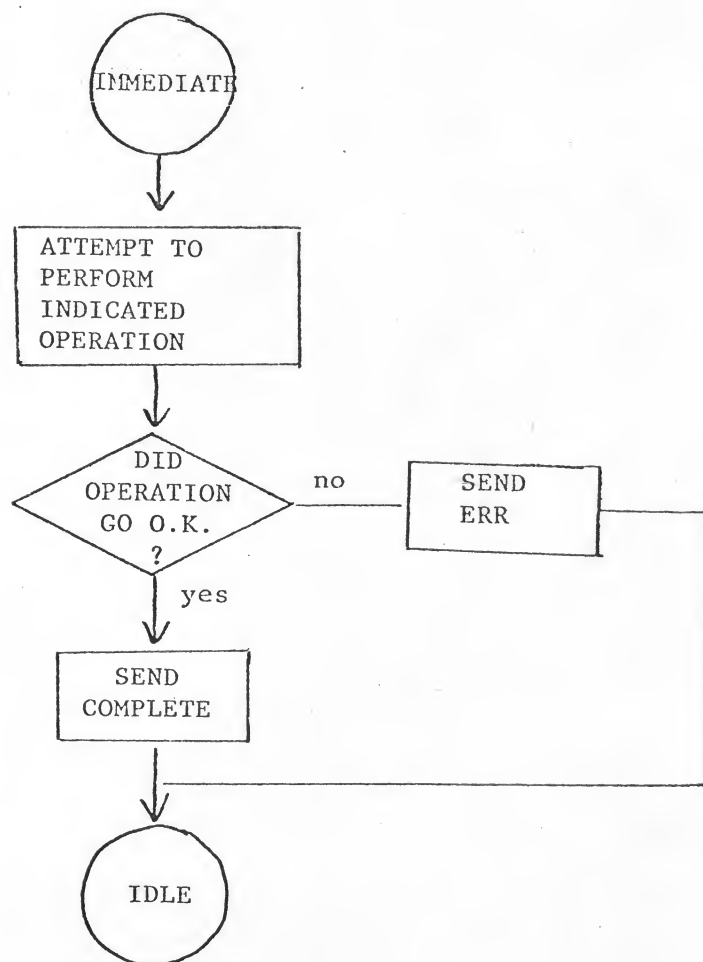

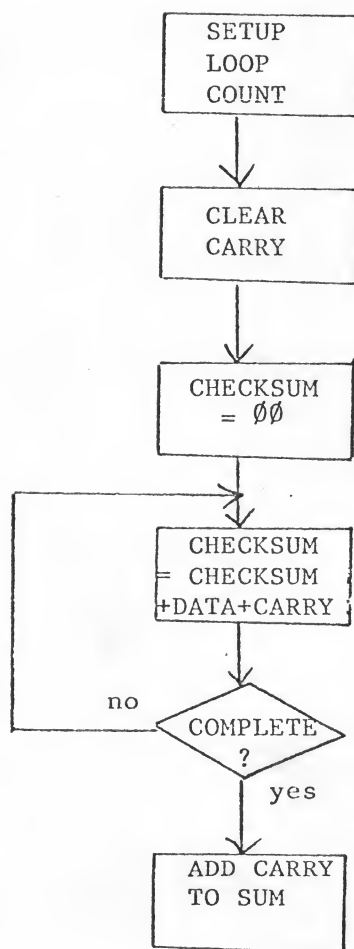See appendix J for existing Serial Bus Device Id's (addresses)

X - 4

The flowchart at the bottom of this page shows a typical device controller's processing of a data frame to be sent to the central processor.

The flowchart at the bottom of this page shows a typical
device controller's of an "immediate' type of command
- no data frame.

```
        ⭘ IMMEDIATE
          │
          ▼
    ┌──────────────┐
    │ ATTEMPT TO   │
    │ PERFORM      │
    │ INDICATED    │
    │ OPERATION    │
    └──────────────┘
          │
          ▼
       ◇ DID
     OPERATION      no    ┌──────────┐
      GO O.K.    ──────── │  SEND    │
         ?              │  ERR     │
          │               └──────────┘
          │ yes                │
          ▼                    │
    ┌──────────────┐           │
    │ SEND         │           │
    │ COMPLETE     │           │
    └──────────────┘           │
          │                    │
          ▼                    │
        ⭘ IDLE ◄──────────────┘
```
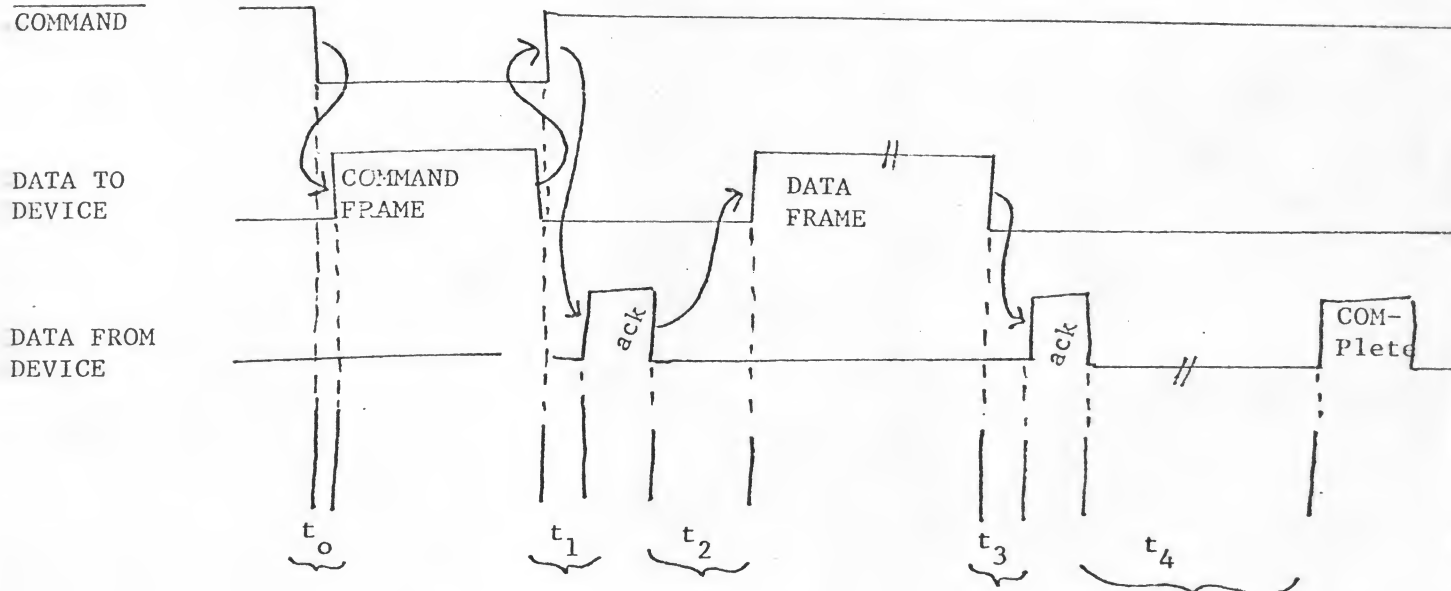
All checksums used as part of the serial bus protocol are simple 8-bit arithmetic sums using the carry bit, as shown below:

```
       ┌──────────────┐
       │    SETUP     │
       │    LOOP      │
       │    COUNT     │
       └──────┬───────┘
              │
              ▼
       ┌──────────────┐
       │    CLEAR     │
       │    CARRY     │
       └──────┬───────┘
              │
              ▼
       ┌──────────────┐
       │   CHECKSUM   │
       │   = ØØ       │
       └──────┬───────┘
              │
     ┌────────▼───────┐
     │ │  CHECKSUM    │
     │ │= CHECKSUM    │
     │ │+DATA+CARRY   │
     │ └──────┬───────┘
     │        │
     │        ▼
     │    ╱────────╲
  no │   ╱ COMPLETE ╲
     └──◀    ?       ▶
         ╲         ╱
          ╲───┬───╱
              │ yes
              ▼
       ┌──────────────┐
       │  ADD  CARRY  │
       │  TO  SUM     │
       └──────────────┘
```

X - 8

3. Bus Timing

The serial Bus Protocol accomodates three basic types of commands: 1) data send, 2) data receive, and 3) immediate (no data frame). A timing diagram is given below for each type:
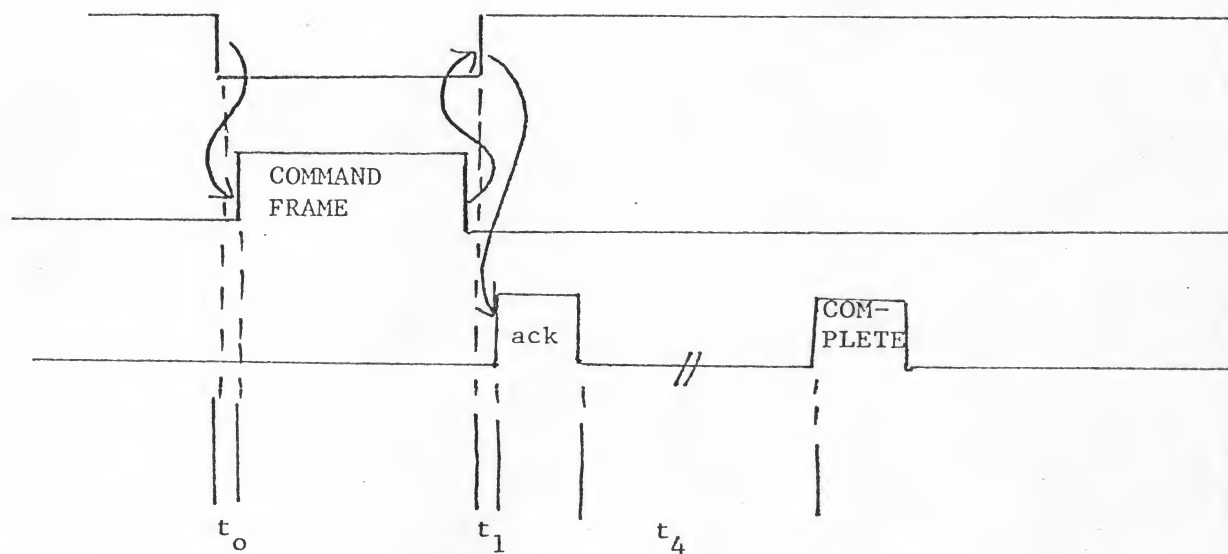
DATA SEND

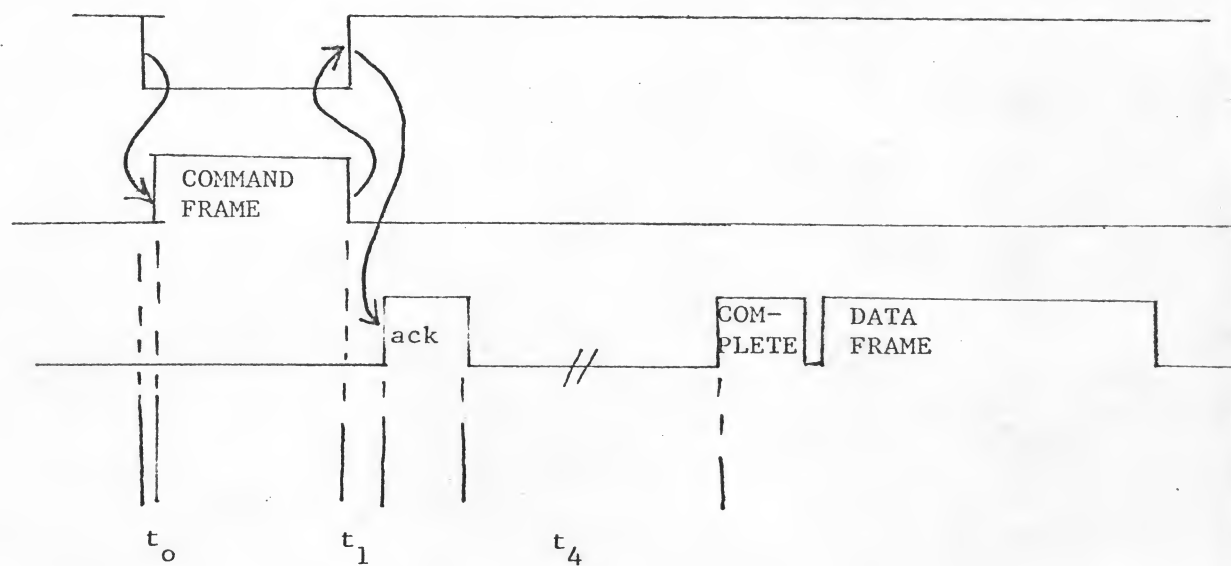|	
to (command frame initiate delay)	= 100 usec – 500 usec
t_1 (command frame acknowledge delay)	= $\emptyset$ – 500 usec
t_2 (controller data receive setup time)	= $25\emptyset$ u – 1 sec.
t_3 (data frame acknowledge delay)	= 150 u – 15 msec.
t_4 (operation complete)	= 150 u – 00

X – 9

IMMEDIATE

COMMAND

DATA TO
DEVICE

COMMAND
FRAME

DATA FROM
DEVICE

ack

COM-
PLETE

t_o t_1 t_4

DATA RECEIVE

COMMAND

DATA TO
DEVICE

COMMAND
FRAME

DATA FROM
DEVICE

ack

COM-
PLETE

DATA
FRAME

t_o t_1 t_4

X - 10

14

4.   Current Devices

This section provides information on the two current intelligent serial bus devices, the 40 column printer and the floppy disk.

4.1   Printer (I.D. = 40_{16})

The printer controller recognizes two commands:
WRITE and STATUS REQUEST.

4.1.1 The printer WRITE command frame contains the following values:

$$Command = 57_{16}$$

Auxilliary Byte 1 = print option

Auxilliary Byte 2 = undefined (not used)

Auxilliary Byte 1 (AUX1) will contain one of the following values –

$$4E_{16} = Normal\ print\ (40\ chars/line)$$
$$53_{16} = Sideways\ print\ (16\ chars/line)$$

The printer WRITE data frame contains either 29 data bytes plus a checksum byte, or 40 data bytes plus a checksum bytes, depending upon the value of AUX1 in the command frame.
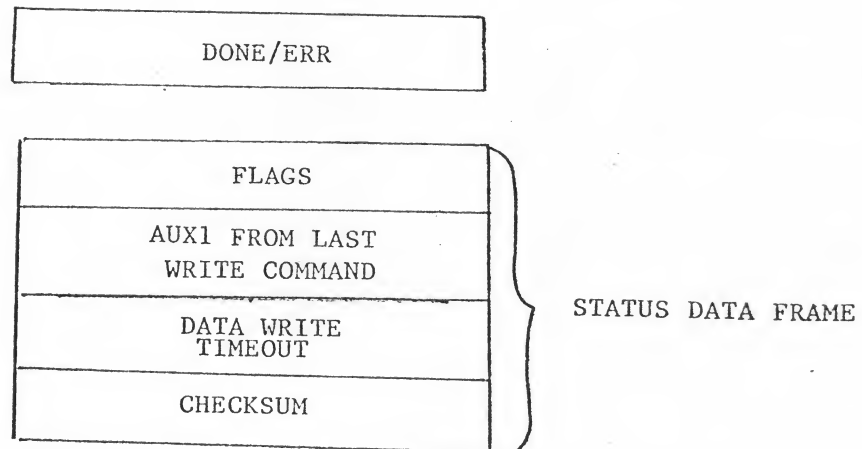
X - 11

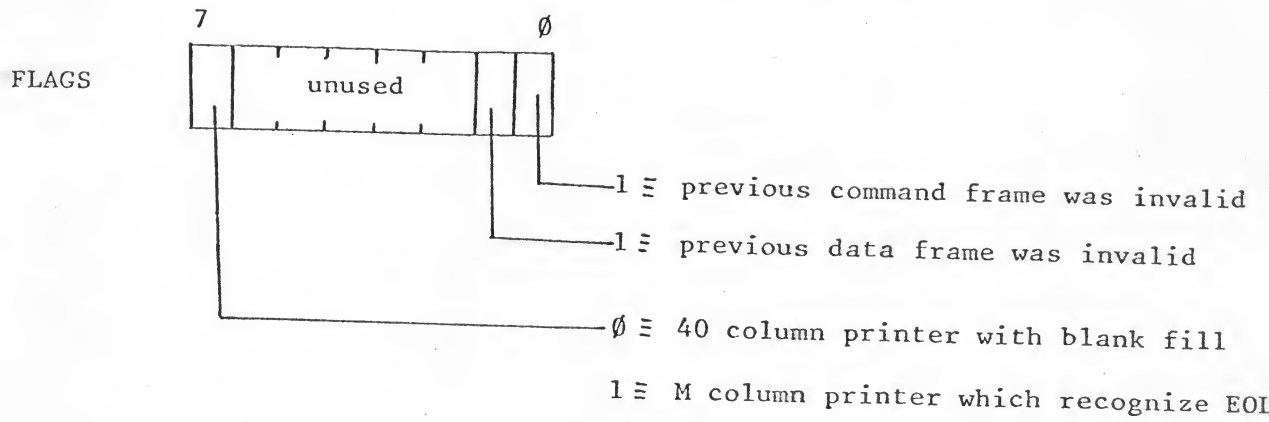4.1.2   The printer STATUS REQUEST command frame contains the
following values:

Command = 53_{16}

Auxilliary Bytes = undefined (not used)

The printer controller will return the status as a
valid data frame as shown below.

DONE/ERR

FLAGS
AUX1 FROM LAST WRITE COMMAND
DATA WRITE TIMEOUT
CHECKSUM

STATUS DATA FRAME

The FLAGS byte contains information relating to the most
recent command prior to the STATUS REQUEST and some controller
constants. The FLAG byte is formatted as shown.

X - 12

FLAGS

```
      7                               Ø
   ┌────┬─┬─┬─┬─┬─┬──┬──┐
   │    │   unused     │  │  │
   └────┴─┴─┴─┴─┴─┴──┴──┘
```

1 ≡ previous command frame was invalid

1 ≡ previous data frame was invalid

Ø ≡ 40 column printer with blank fill

1 ≡ M column printer which recognize EOl

Data Write Timeout = maximum time to print a line of data assuming worst case controller produced cool-off delay. (1   sec.)

X – 13

4.2   Floppy Disk Controller (I.D.s = $31\text{-}34_{16}$)

The floppy disk controller recognizes   commands:
READ, PUT, WRITE, FORMAT, and STATUS REQUEST.

4.2.1   The disk READ command frame contains the
following values:

> Command   = 52_{16}
> Auxilliary Byte 1 = Disk Sector L.S.B. $(1 - 72\emptyset)$
> Auxilliary Byte 2 = Disk Sector M.S.B.

The disk READ data frame contains 128 bytes of data
plus a checksum byte.

4.2.2   The disk PUT command frame contains the
following values:

> Command   = 50_{16}
> Auxilliary Byte 1 = Disk Sector L.S.B.
> $(1\text{-}720)$
> Auxilliary Byte 2 = Disk Sector M.S.B.

4.2.3   The disk WRITE command (read after write check) frame
contains the following values:

> Command   = 57_{16}
> Auxilliary Byte 1 = Disk Sector L.S.B.
> $(1\text{-}720)$
> Auxilliary Byte 2 = Disk Sector M.S.B.
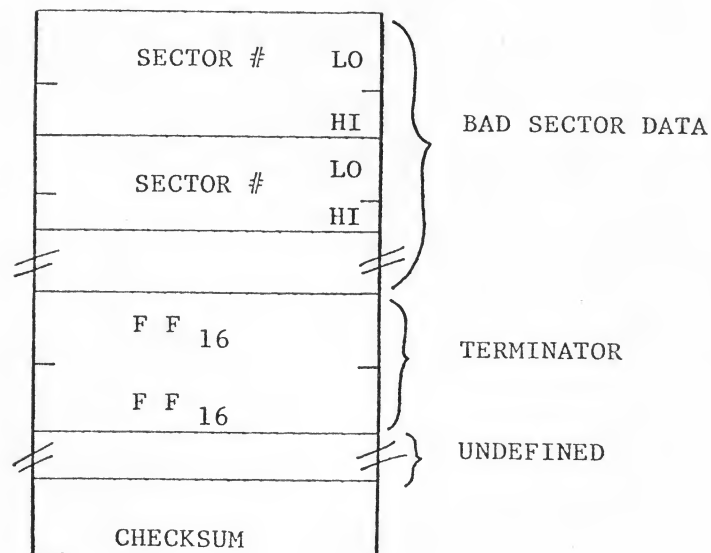
X - 14

The disk WRITE data frame contains 128 bytes
of data plus a checksum byte.

4.2.4   The disk FORMAT command frame contains the following
values:

Command   = 21_{16}
Auxilliary Bytes = undefined (unused)

The disk FORMAT data frame (returned by the controller)
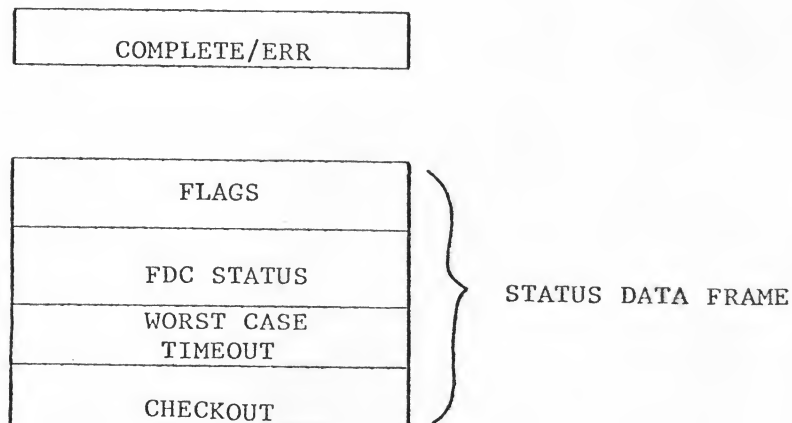contains 128 bytes of data plus a checksum byte, formatted
as shown below:

The disk is formatted and then verified by the disk
controller; the FORMAT data frame contains from zero
to 63 bad-sector numbers, followed by two bytes of all
ones, followed by fill bytes   to pad
out the data block to 128 bytes, followed by a checksum.
The sector numbers may be in any order.

4.2.5   The disk STATUS REQUEST command frame contains the
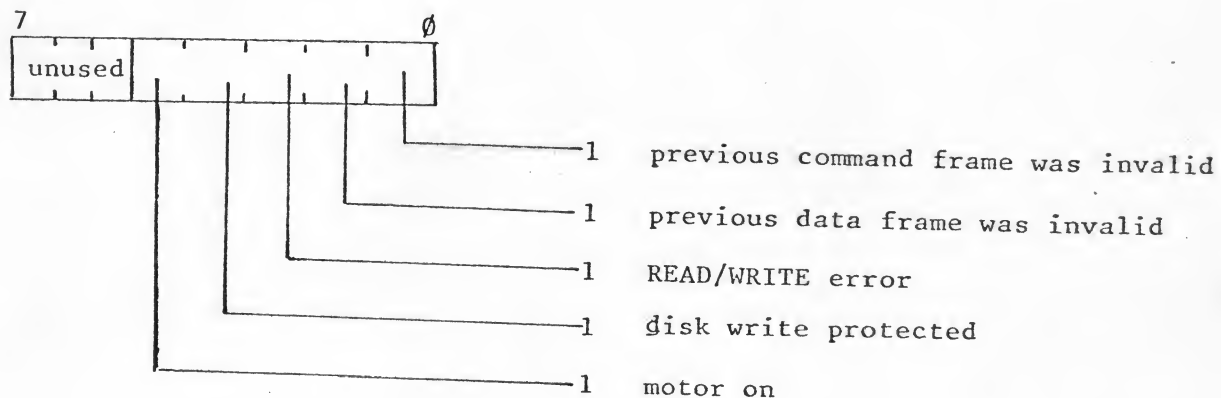following values:

Command = 53_{16}

Auxilliary Bytes = undefined (unused)

The disk controller will return the status as a
valid data frame as shown below.

COMPLETE/ERR

FLAGS
FDC STATUS
WORST CASE TIMEOUT
CHECKOUT

STATUS DATA FRAME

X – 16

The FLAGS byte contains information relating to
the most command prior to the STATUE REQUEST, and
is formatted as shown.

Worst case timeout   =   maximum time to wait for
                             slowest operation (FORMAT).
                             (1 sec.)

Appendix K - Serial Bus Command Codes/Control Codes

Commands	Hex	ASCll
PUT	50	('P')
READ	52	('R')
WRITE	57	('W')
STATUS	53	('S')
FORMAT	21	('!')

Control Codes	Hex	ASCll
ACK	41	('A')
NAK	4E	('N')
COMPLETE	43	('C')
ERR	45	('E')

Appendix J - Serial Bus Device Address

Floppy Disks $31 - 34_{16}$

Printer 40_{16}

Modem $50 - 58_{16}$